In re Application of: Eli YANOVSKY
Serial No.: 10/520,274
Filed: January 18, 2005
Office Action Mailing Date: August 4, 2009

Examiner: KANAAN Simon P.
Group Art Unit: 2432
Attorney Docket: 29238

REMARKS

Reconsideration of the above-identified application in view of the amendments above and the remarks following is respectfully requested.

Claims 1 - 48 are in this Application. Claims 1 – 48 have been rejected under 35 U.S.C. § 103. Claims 1, 21 and 37 have been amended herewith.

35 U.S.C. § 103 Rejections

Claims 1 – 48 have been rejected for lack of inventive step.

Claims 1, 21 and 37 were amended in the previous responses to stress the point that the *encryption keys* are *generated separately* at the two different parties, using separate randomizers, which have identical settings.

Applicant now amends the claims to define that the two randomizers, one at each party, and both set with identical settings, are such that so the *settings* define the *same derived sequence* from the *exchanged bitstream*, thus:

> "a random selector configured with selection settings identical to those at said second party said selection settings defining a selection, from said bitstream, of a series of bits in accordance with a randomization within said random selector, said randomization seeded by said data exchanged between said parties, said randomization being identical to a randomization carried out at said second party, thereby ensuring that said series of bits is identically selected at both parties;"

Seheidt does not teach a randomizer, as the Examiner concedes at page 3 third line from the end, of his Office Action.

Seheidt further does not teach

> a key generator configured for separately generating at said first party a key for encryption/decryption based on said series of bits,

since the series of bits is that derived using the settings, and Seheit obtains the key from *two key parts exchanged* between the parties, not from *a series of bits obtained, via settings, from a data stream,* contrary to the requirements of the claim.

Examiner cites Maurer, in order to teach the randomizer.

Examiner points to Maurer Fig. 1 and claims that this teaches:

"key index generator sends signal to two different parties which have their own randomizers which create identical keys. One party uses the key to encrypt and send the message. The second party receives the message and decrypts it with its own generated key." (Office Action end of first full paragraph on page 3).

Applicant respectfully points out that this is not a correct interpretation of Fig. 1 of Maurer and the corresponding description.

First of all the key index generator 18 of Maurer does not "send signal to two different parties which have their own randomizers which create identical keys", contrary to the contention of the Examiner. The output of key index generator merely performs an *index* function on an *already stored* library of key material memory 16, and 36(of the transmitter and the receiver, accordingly) in order to provide a dynamically changing key.

The key index function does *not* generate a key but merely *varies* locations within an *existing stored* library of key material, *dynamically*.

The function of *varying* locations within a stored library of key material *dynamically* at both parties is taught by Maurer, but this is a far cry from generating identical keys randomly and not from fixed material at both parties.

The Examiner is referred to Maurer Abstract: “the key is selected from a library of key material according to a key index signal. The key index signal, which is transmitted to all station that must decrypt the message signal....”. The examiner is also referred to Maurer column 3 line 50: “stored key 14 is made up of elements selected from memory 16”; and to column 3 line 55: “stored key 34 is selected from memory 36 according to the key index signal from generator 18”. The description of Maurer is very clear that *libraries of key material are stored at each party and one party send to the other via communication line the key index signal for selecting from the library the current key to use.* Furthermore, there is *no independent generation of keys* at the two parties.

What Maurer does is dynamically *vary* locations within the fix long-term stored library of key material (16 and 36) at the two parties. However Maurer requires the sending of the key index signal from one party to the other; thus, both the fix long-term stored library of key material and the sending of the key index signal from one party to the other, are precisely the problem that the present application was intended to avoid.

Thus if Maurer were to be combined with Seiheit, it would *still be necessary* to provide an *initially stored* library of key material 16and 36 at the two parties.

Hence the combination of Maurer with Seiheit does not teach or hint at the present invention but merely begs the problem that the present invention was provided to solve.

The combination of Maurer and Seiheit thus fails to teach

> "a key generator configured for separately generating at said first party a key for encryption/decryption based on said series of bits",

since Maurer merely points to different locations in the stored library of key material . Seiheit uses exchanged key halves, and not the series of bits defined in the claim, as explained above.

The same amendment is made to the remaining independent claims.

The dependent claims are believed to be allowable as being dependent on an allowable main claim.

In view of the above amendments and remarks it is respectfully submitted that claims 1 - 48 are now in condition for allowance. A prompt notice of allowance is respectfully and earnestly solicited.

The inventor wishes to add the following comments:

The present application deals with the deficiencies of modern data security systems & methods to give a thorough answer to the Key Management issue, especially to the challenge of the need to refresh and change the secret key at high rate, simultaneously at both conversing parties, who are remote from each other, and wherein the communication between them is carried out over open and insecure lines.

The challenge is magnified in the modern era since the methods used by adversaries, hackers and pirates to penetrate security systems are sophisticated. It remains true that no matter how advanced the system, an adversary can break in by stealing the key, and techniques for doing this include Phishing, Trojan horses, RAT, and so on, and even something as basic and old fashioned as bribing an insider.

Thus it is understood that nowadays keys must be changed and refreshed at an accelerated rate, and one cannot merely count on the infeasibility of breaking the secret key over time. Solutions exist for regular changing of keys, for example a fixed long term library of keys or bits to use for generating keys, or a top secret fixed long term master key that is used either to refresh and generate working data keys, or to encrypt fresh data keys along their transfer from one party where it was generated to the other conversing party, or a generator that works in an advancing or circular manner using a fixed long term secret key, or a fixed long term library of keys or bits to use for generating keys and so on. All of these fail if the adversary manages to find the fixed ultra secret key or obtain the library.

As well as the key management issue there is an authentication issue.

The challenge is therefore how to generate secret keys on an on-going basis, not by a deterministic manner nor under a long-term secret key, nor by a fixed long term library of keys or bits to use for generating keys, but *randomly each time* and *simultaneously at both remote parties*, *without* sending over the lines secret data or keys, even if secured (encrypted) by a long-term secret key or by using a fixed long term library of keys or bits to use for generating keys (master or public/private method).

The above issues and challenges, discussions and prior art methods are lengthily discussed and regarded to in our patent application.

The present embodiments provide a solution to the above as follows:

The two conversing parties start by having the same inside secret information (data). From that secret information they may, using identical but random processes, generate the first secret key to be used in the first segment of their coming secret conversation over the open & insecure lines.

Thus whereas in the prior art an initial key or segment therefore or encrypted key is transmitted, in the present invention all that is transmitted is a stream of data which seeds a randomization process, so that even if an eavesdropper were to intercept the data he would not be able to derive the initial key.

Continuing with the conversation, and at the first conversation segment both parties use the communicated ciphertext segment itself as multi sources of randomness. Both parties may then use their same inside secret data, to have or generate in the same secret manner a way to select a same source of randomness from the multiple sources of randomness available. In one embodiment of the invention, addresses are generated to define places and an order of bits to pick out from the ciphertext. The bits make up a random sequence of bits that change in a random manner the same inside secret information to a new randomly generated sequence.

For the next communicated segment, the new inside secret information is used to generate a new secret key to be used at this segment, and a new secret manner of choosing (new addresses) how to identically select a same source of randomness from the new ciphertext segment. The process continues over successive segments of the exchange.

Thus nothing is fixed long term, and the relationship between any one stage and its successor is purely random.

The present application further deals with synchronization challenges, with a requirement neither to count on fixed & long-term secrets nor to send secret data over the lines, nor to rely on any kind of library.

Regarding the citations:

1) US Patent No. 5,375,169 by Seheidt et al, is a simplified method known as the Diffie - Hellman method to create a secret key between two parties who have no secret (key) between them. In Diffie Hellman, each party sends half of the secret (key) to the other party, so that both can create the full secret key. Seheidt et al, is

however, more vulnerable than pure Diffie - Hellman method, as the half keys he exchanges may be formed into a key as they are. In pure Diffie – Hellman, the pieces are sent in a mathematical format that no eavesdropper to the line can obtain the correct format of the combination to feasibly create the secret key.

Even though the Diffie - Hellman Method is a mathematical way to send halves of a secret in a way that is unfeasible for third parties to create the full secret, it does not teach the present invention but rather deals with the mathematics of key generation itself and how to camouflage sent secrets. Furthermore, Diffie - Hellman lacks authentication needs, and specifically is exposed to the man in the middle attack. Diffie Hellman's only solution to this is to provide the halves of the secrets to a public library. Such a response is not a real solution for modern challenges.

On this ground US5,375,169 by Seheidt et al, is less strong.

Seheidt et al deals with the Key Management issue, but does so by openly and insecurely sending halves of the key, coupled with an error detecting code, and if the halves are received correctly according to the error detecting code then they are combined to create the key for the message to communicate 'secretly', and if not correct – inhibit communication.

2) US Patent No. 5,253,294 by Maurer, is a patent for creating a data key for a message by using a long term secret LIBRARY of key material (numbers/bits), being held at both parties, and at any message start, (and at every day) the sender generates a "key index signal" (a random number) to form the library addresses for key selection from the library to be used by both parties. An index to the library is the key index signal which is sent to the receiver encrypted by the "day key'. The process is cascaded to provide a dynamically changing key.

In Maurer, the library is prestored, so that there is no key generation at all. The whole system is exposed to anyone who gets hold of the library, and at least there is a need to send by a secured manner the key index signal.

3) US Patent No. 5,923,758 by Khamharn et al, merely deals with how to regain synchronization at a receiver of a 'keyless system'.

Even though those patents may use similar terms, such as key management, random selecting, resynchronization, etc., this is where all similarity ends.

In view of the above amendments and remarks it is respectfully submitted that claims 1-48 are now in condition for allowance. A prompt notice of allowance is respectfully and earnestly solicited.

Respectfully submitted,

Martin D. Moynihan

Martin D. Moynihan
Registration No. 40,338

Date: January 4, 2010

Enclosure:

- Petition for Extension (Two Months)